POINTS OF SIGNIFICANCE

Classification and regression trees

Decision trees are a simple but powerful prediction method.

We have seen how a categorical or continuous variable can be predicted from one or more predictor variables using logistic[1] and linear regression[2], respectively. This month we'll look at classification and regression trees (CART), a simple but powerful approach to prediction[3]. Unlike logistic and linear regression, CART does not develop a prediction equation. Instead, data are partitioned along the predictor axes into subsets with homogeneous values of the dependent variable—a process represented by a decision tree that can be used to make predictions from new observations.

We'll begin with a simple example of one continuous predictor variable, X, and one categorical dependent variable, Y—these variables could be the level of expression of a gene and one of three eye color categories, respectively. Using a sample with $n = 60$ points equally distributed among three categories (**Fig. 1a**), let's build a decision tree classifier that predicts the color category Y based on the value of X.

To do this, we'll iteratively split X into intervals that are as homogeneous as possible with respect to Y. In the first iteration, the sample is split into two subsets by considering all possible partitions $X < x$ and $X > x$, where x is the midpoint between two adjacent observed values of X. We select x to maximize the information gain $IG(S_1,S_2) = I(S) - n_1I(S_1)/n - n_2I(S_2)/n$, which measures how well the classes are separated by the split of set S into subsets S_1 and S_2 with n_1 and n_2 points. By maximizing $IG(S_1,S_2)$ we favor splits into subsets that are homogeneous with respect to categories and heterogeneous with respect to one another.

The definition of $IG(S_1,S_2)$ depends on the impurity function $I(S)$, which measures class mixing in a subset. For classification trees, a common impurity metric is the Gini index, $I_g(S) = \sum p_i(1 - p_i)$, where p_i is the fraction of data points of class i in a subset S. The Gini index is minimum ($I_g = 0$) if the subset comprises a single class and maximum ($I_g = (k - 1)/k$) when k classes are evenly represented. **Figure 1a** shows the Gini index for the full sample $I_g(S)$ and for each subset ($I_g(S_1)$, $I_g(S_2)$) in three possible splits. For example, for the boundary at $X = 20$, the p_i values for the left subset are (17/20, 3/20, 0) giving $I_g(S_1) = 0.26$ and for the right subset (3/40, 17/40, 20/40) giving $I_g(S_2) = 0.56$. The Gini index information gain is therefore $IG_g = 0.66 - 20/60 \times 0.26 + 40/60 \times 0.56 = 0.21$. When we calculate IG_g for every possible split, we find the maximum $IG_g = 0.25$ at $X = 38$ (**Fig. 1b**).

Two other common impurity metrics are entropy, $I_e(S) = -\sum p_i \log_2(p_i)$ (relatively slow to compute because of the $\log_2$), and misclassification error, $I_c(S) = 1 - \max(p_i)$. The information gain functions based on the Gini index and entropy behave similarly—for our sample, both reach a unique maximum, though at slightly different positions (**Fig. 1b**). The misclassification error is simply the fraction of points in a subset that aren't in the majority vote class. This error is useful for pruning a decision tree but less so for growing one

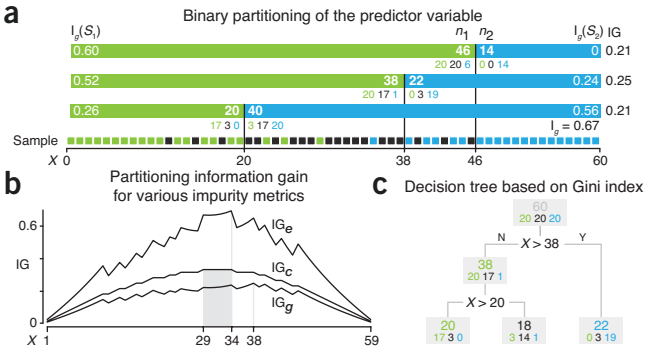

Figure 1 | A classification decision tree is built by partitioning the predictor variable to reduce class mixing at each split. (**a**) An $n = 60$ sample with one predictor variable (X) and each point belonging to one of three classes (green, dark gray, blue). Three possible splits are shown at $X = 20$ ($X < 20$ and $X > 20$), $X = 38$ and $X = 46$ along with the number of points in the resulting subsets (n_1, n_2), their breakdown by class (colored numbers), the purity of the subset ($I_g(S_1)$, $I_g(S_2)$) and information gain for the split (IG_g), based on the Gini index. The sample's Gini index is $Ig = 0.67$. (**b**) The information gain based on Gini index (IG_g), entropy (IG_e) and misclassification error (IG_c) for all possible first splits. The maxima of IG_g, IG_e and IG_c are at $X = 38$, 34 and 29–34, respectively. (**c**) The decision tree classifier of sample in **a** based on IG_g. Large text in each node is the number of points colored by predicted class. Smaller text indicates class membership in each subset. N, no; Y, yes.

because it is less sensitive to class distribution—for our sample, IG_c does not have a unique maximum (**Fig. 1b**). For example, a sample with class membership (20,20,20) split into (20,10,5) and (0,10,15) gives $IG_g = 0.13$. Splitting instead into (20,20,5) and (0,0,15) gives $IG_g = 0.22$, which is higher and preferred. However, if we used the misclassification error, we would obtain $IG_c = 0.25$ for both splits and could not decide between them. Arguably, the second split is 'purer', since it generates a subset that contains only one class.

Once the first split is chosen, each of the two subsets is split again using the same approach, and the process continues iteratively. The splits generate a decision tree whose nodes correspond to the subsets of the data and whose branches correspond to partitioning of the variable above or below a splitting value for a predictor. Each node is associated with the class that appears most often in the subset, if the cost of misclassification of a data point is independent of class. In **Figure 1c** we show the full decision tree that classifies our sample based on Gini index—the data are partitioned at $X = 20$ and 38, and the tree has an accuracy of 50/60 = 83%. Interpreting the decision tree in the context of our biological example, we would associate observations at expression level $X < 20$ with the green color category. The tree based on entropy (not shown) is similar, with accuracy of 47/60 = 78% and boundaries at $X = 25$ and 34.

When the dependent variable is continuous, its value can be predicted using regression trees. These predict using the average values of $\bar{y}$ within each subset, which are selected to minimize the mean square error, $MSE = \sum_i(\bar{y} - y_i)^2/n$. For example, the continuous nonlinear function in **Figure 2a** can be trivially estimated by its average across the full interval, $\bar{y} = 0.58$ with MSE = 0.11. We can lower the MSE by partitioning the interval and minimizing $n_1MSE_1/n + n_2MSE_2/n$, the weighted average of MSE of both subsets. Doing this, we find that the boundary at $X = 40$ yields the lowest value (**Fig. 2b**), and this split forms the first branching in the regression tree (**Fig. 2c**). As before, the nodes are subsets but are now associated with the average value of the dependent

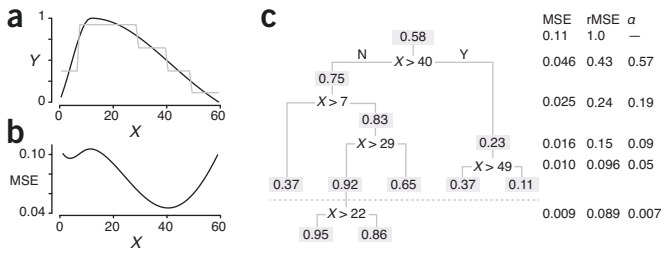

Figure 2 | Regression trees predict a continuous variable using steps in which the prediction is constant. (**a**) A nonlinear function (black) with its prediction (gray) based on a regression tree. (**b**) Splits in the regression tree minimize mean square error (MSE), which is shown here for all possible positions of the first split. (**c**) The full regression tree for the prediction shown in **a**. For each split, the absolute MSE and relative (to the first node) rMSE is shown along with the difference in successive rMSE values, α. The tree was built with cutoff of $\alpha = 0.01$, which terminates the growth of the tree at the dashed line.

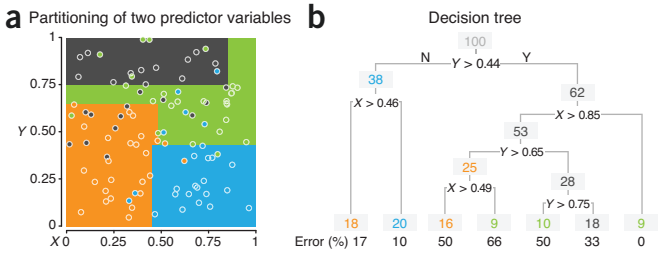

Figure 3 | Decision trees can be applied to many predictor variables. (**a**) Classification boundaries for an $n = 100$ sample with two predictor variables (X, Y) and four categories (colors). (**b**) Decision tree built with $\alpha = 0.01$ of the data set in **a**.

variable of the points in the subset; 0.75 for $X \leq 40$ and 0.23 for $X > 40$. As we split our regression tree, we get to a state with five leaves and MSE = 0.010 (**Fig. 2c**).

We can always continue splitting until we build a tree that is 100% accurate, except where points with the same predictors have different classes (e.g., two observations with same gene expression belong to different color categories). However, this would almost always overfit the data (e.g., grow the tree based on noise) and create a classifier that would not generalize well to new data[4]. To determine whether we should continue splitting, we can use some combination of (i) minimum number of points in a node, (ii) purity or error threshold of a node, or (iii) maximum depth of tree.

For both classification and regression, a useful stopping criterion is to require that each split improves the relative error by at least α, a predetermined value of the complexity parameter. This parameter acts to regularize the cost function of growing the tree[5] by balancing the cost with a penalty for adding additional partitions. For example, as we grow our regression tree, we monitor the relative MSE (rMSE) of each split and the amount of decrease α at each split (**Fig. 2c**). Splitting at $X = 49$ improves the rMSE by $\alpha = 0.05$. However, the next candidate split at $X = 22$ lowers rMSE by only $\alpha = 0.007$. If we use a cutoff of $\alpha = 0.01$, this split would not be accepted, and tree growth would end.

An alternative to limiting tree growth is pruning using k-fold cross-validation. First, we build a reference tree on the entire data set and allow this tree to grow as large as possible. Next, we divide the input data set into training and test sets in k different ways to generate different trees. We evaluate each tree on the test set as a function of size, choose the smallest size that meets our requirements and prune the reference tree to this size by sequentially dropping the nodes that contribute least.

In general, trees can be built using forward (tree-growing) or backward (tree-trimming) algorithms. The original CART used tree trimming because the splitting algorithm is greedy and cannot foresee better splits ahead, while trimming grows the whole tree so that the value of the splits can be assessed at the end of the process.

Decision trees can be applied to multiple predictor variables—the process is the same, except at each split we now consider all possible boundaries of all predictors. **Figure 3** shows how a decision tree can be used for classification with two predictor variables.

If the data set and the number of predictor variables is large, it's possible to encounter data points that have missing values for some predictor variables. This can be handled by filling in these missing values based on surrogate variables selected to split similarly to the selected predictor.

The creation of the tree can be supplemented using a loss matrix, which defines the cost of misclassification if this varies among classes. For example, in classifying cancer cases it may be more costly to misclassify aggressive tumors as benign than to misclassify slow-growing tumors as aggressive. The penalty is applied as a weight to the impurity index. The node is then assigned to the class that provides the smallest weighted misclassification error. In our example, we did not differentially penalize the classifier for misclassifying specific classes.

Decision trees are very effective and are readily interpreted. However, individual trees can be very sensitive to minor changes in the data, and even better prediction can be achieved by exploiting this variability to grow multiple trees from the same data. This will be the topic of the next column.

Corrected after print 28 July 2017.

COMPETING FINANCIAL INTERESTS
The authors declare no competing financial interests.

Martin Krzywinski & Naomi Altman

1. Lever, J., Krzywinski, M. & Altman, N. *Nat. Methods* **13**, 541–542 (2016).
2. Altman, N. & Krzywinski, M. *Nat. Methods* **12**, 999–1000 (2015).
3. Breiman, L., Friedman, J.H., Olshen, R.A. & Stone, C.J. *Classification and Regression Trees* (Wadsworth, 1984).
4. Lever, J., Krzywinski, M. & Altman, N. *Nat. Methods* **13**, 703–704 (2016).
5. Lever, J., Krzywinski, M. & Altman, N. *Nat. Methods* **13**, 803–804 (2016).

Martin Krzywinski is a staff scientist at Canada's Michael Smith Genome Sciences Centre. Naomi Altman is a Professor of Statistics at the Pennsylvania State University.

Corrigendum: Classification and regression trees

Martin Krzywinksi & Naomi Altman
Nat. Methods **14**, 757–758 (2017); **published online 28 July 2017; corrected after print 28 July 2017**

In the version of this article initially published, the expression (g_1, g_2) used to describe a sample subset in the **Figure 1** legend was incorrect. The correct expression is $(I_g(S_1), I_g(S_2))$. The error has been corrected in the HTML and PDF versions of the article.